

*Exceptional service in the national interest*



# Extending van Leer's Algorithm to Multiple Dimensions

S. J. Mosso<sup>1</sup>, T.E. Voth<sup>2</sup>, and R.R. Drake<sup>3</sup>

Sandia National Laboratories

SAND 2013-7261C

<sup>1</sup> [sjmosso@sandia.gov](mailto:sjmosso@sandia.gov)

<sup>2</sup> [tevoth@sandia.gov](mailto:tevoth@sandia.gov)

<sup>3</sup> [rrdrake@sandia.gov](mailto:rrdrake@sandia.gov)

# Introduction:

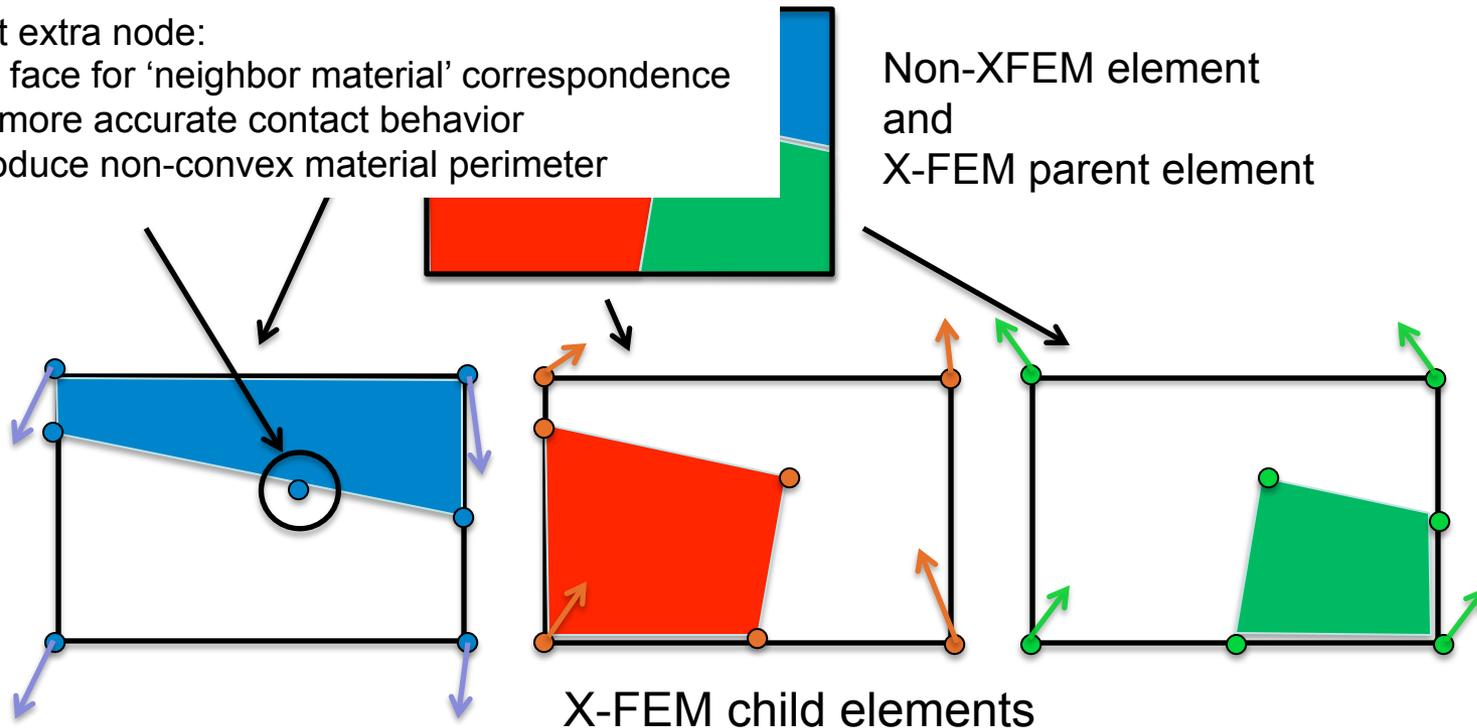
- Overview of X-FEM algorithms for interface reconstruction and remapping
- Development of the van Leer interpolation for remap
- Demonstrate with a verification problem
- Conclusion and future directions

# X-FEM Overview:

- **X-FEM is part of Alegra**
  - **Selected through input deck**
  - **Voth's clever implementation allows hydro/strength algorithms to be shared**

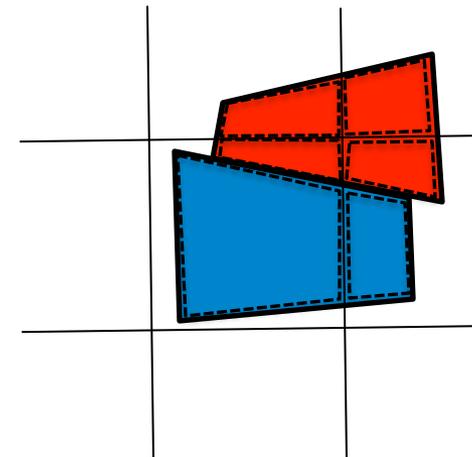
Important extra node:

- divides face for 'neighbor material' correspondence
- allows more accurate contact behavior
- can produce non-convex material perimeter



# Remapping algorithm:

- 2D Design choices: swept face or intersection remap
- Swept face
  - Originally unclear how to implement
  - Due to two directional sweeps (2D), increased dissipation
  - Possible lower fidelity due to approximations
- Intersection remap:
  - Based upon Dukowicz<sup>1</sup> and Ramshaw<sup>2</sup> (without node coordinate randomization)
  - Much more complex algorithm
  - Each material is remapped individually
  - Able to remap non-convex material polygons
  - More expensive than swept remap



Remapping two material donor element onto its neighbors

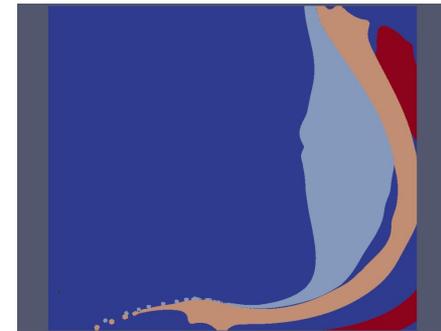
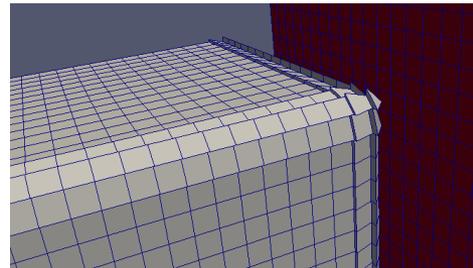
[1] J.K. Dukowicz, Conservative Rezoning (remapping) for general quadrilateral meshes, J. Comput. Phys., 54 (1984), pp. 411 – 412,

[2] J.D. Ramshaw, Conservative Rezoning Algorithm for generalized Two-Dimensional Meshes, J. Comput. Phys, 59 (1985), pp. 193-199.

# Interface Reconstruction:

- Patterned Interface Reconstruction (PIR)
  - Based upon Youngs' algorithm
  - PIR works in physical space (as opposed to unit-cube)
  - Provides a conservative material description
  - Keeps 'neighboring material' connections for the contact
  - Manual priorities with enhanced material placement (materials aren't required to use onion-skin).
  - Automatic priorities (see R. Kramer's talk)
  - Polygonal element and material description guarantees non-crossing interfaces
  - Interface smoothing is rarely used
  - Adds interface segments (2D) or polygons (3D) to give  $C_0$  continuous interface description for contact and DSD

Reconstructed  
interfaces of 3D bar  
showing addition of  $C_0$   
polygons (skinning)



Whipple Shield  
calculation with Alegra  
X-FEM illustrating  
multiple occurrences of  
void material in fixed  
priority order  
reconstruction

# Multiple Dimensional MUSCL algorithm

- Many similar efforts at extending MUSCL have occurred
- MUSCL<sup>1</sup> concepts:
  - Uses a Taylor Series expansion to represent the distribution of conserved scalar quantities over a donor element

$$F_{ext} = F_{home} + (x - x_c) \frac{\partial F}{\partial x} + (y - y_c) \frac{\partial F}{\partial y} + \frac{1}{2} (x - x_c)^2 \frac{\partial^2 F}{\partial x^2} + (x - x_c)(y - y_c) \frac{\partial^2 F}{\partial x \partial y} + \frac{1}{2} (y - y_c)^2 \frac{\partial^2 F}{\partial y^2}$$

- Commonly uses only 2<sup>nd</sup> order terms or 3<sup>rd</sup> order Taylor series (all terms above)
- The distribution function is integrated over the intersection elements

[1] B. van Leer, Towards the Ultimate Conservative Difference Scheme, V. A second-order sequel to Godunov's Method, J. Comput. Phys., 32 (1979), pp. 101-136.

# Achieving monotonicity

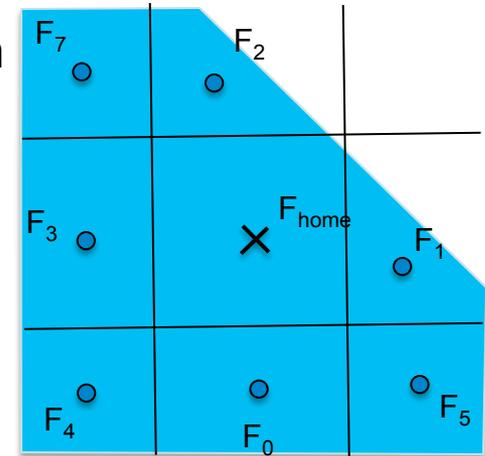
- Monotonicity (create no new maxima or minima in the distribution):

If  $F_{home} \geq F_n : F_{home} \geq F_{ext} \geq F_n$

else  $F_{home} < F_n : F_{home} < F_{ext} < F_n$

this says that the extrapolated function is bounded by the home and neighbor values of  $F$

- When multiple dimensional Taylor series is used, usually the number of neighbors overdetermines the values of the derivatives
- Use an 'optimization' algorithm to achieve a solution.
- Implement the monotonicity conditions as constraints
- The objective function will be how well does the extrapolated value approximate all the neighbors
- We will use a linear programming algorithm to solve the problem in each donor element (lp\_solve 5.5 for now)
- Linear programming evenly weights the neighbors (whereas least-squares more heavily weights the ill-fitted neighbors)



# Algebra of monotonicity

- How are the constraints implemented?

for example:  $F_{home} \geq F_n : F_{home} \geq F_{ext} \geq F_n$

- First constraint for neighbor

$$F_{home} \geq F_{ext}$$

$$F_{home} \geq F_{home} + \Delta x \frac{\partial F}{\partial x} + \Delta y \frac{\partial F}{\partial y} + \frac{1}{2} \Delta x^2 \frac{\partial^2 F}{\partial x^2} + \Delta x \Delta y \frac{\partial^2 F}{\partial x \partial y} + \frac{1}{2} \Delta y^2 \frac{\partial^2 F}{\partial y^2}$$

$$0 \geq \Delta x \frac{\partial F}{\partial x} + \Delta y \frac{\partial F}{\partial y} + \frac{1}{2} \Delta x^2 \frac{\partial^2 F}{\partial x^2} + \Delta x \Delta y \frac{\partial^2 F}{\partial x \partial y} + \frac{1}{2} \Delta y^2 \frac{\partial^2 F}{\partial y^2}$$

- Second constraint for neighbor

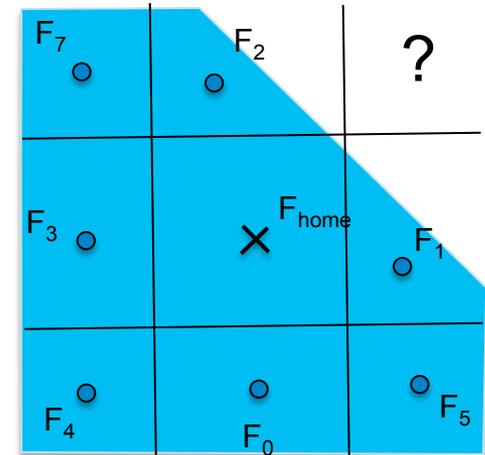
$$F_n \leq F_{ext}$$

$$F_n \leq F_{home} + \Delta x \frac{\partial F}{\partial x} + \Delta y \frac{\partial F}{\partial y} + \frac{1}{2} \Delta x^2 \frac{\partial^2 F}{\partial x^2} + \Delta x \Delta y \frac{\partial^2 F}{\partial x \partial y} + \frac{1}{2} \Delta y^2 \frac{\partial^2 F}{\partial y^2}$$

$$F_n - F_{home} \leq \Delta x \frac{\partial F}{\partial x} + \Delta y \frac{\partial F}{\partial y} + \frac{1}{2} \Delta x^2 \frac{\partial^2 F}{\partial x^2} + \Delta x \Delta y \frac{\partial^2 F}{\partial x \partial y} + \frac{1}{2} \Delta y^2 \frac{\partial^2 F}{\partial y^2}$$

# Unoccupied Neighbors?

- Require four occupied neighbors for 3<sup>rd</sup> order algorithm and three occupied neighbors for 2<sup>nd</sup> order
- If this number of neighbors isn't present, lower algorithm in home element by one order
- If there are enough neighbors, two options (user input):
  - **Allow extrapolated derivatives:** omit constraints for this neighbor and allow the other neighbors to determine the distribution
    - The algorithm can create a new maximum or minimum but it is usually 'reasonable'
  - **Deny extrapolated derivatives** in direction of missing neighbors:
    - Implemented by using value of  $F_{\text{home}}$  for missing  $F_n$
    - In plots it appears as a 'ledge' around the material boundary



# Displacements used in constraints

- Constraints can be ‘excessively limiting’
  - An example is when two ‘opposite’ neighbors have very small differences with  $F_{home}$  and the dot product of the displacements aren’t near 1:

$$F_{home} \geq F_{ext} \geq F_1$$

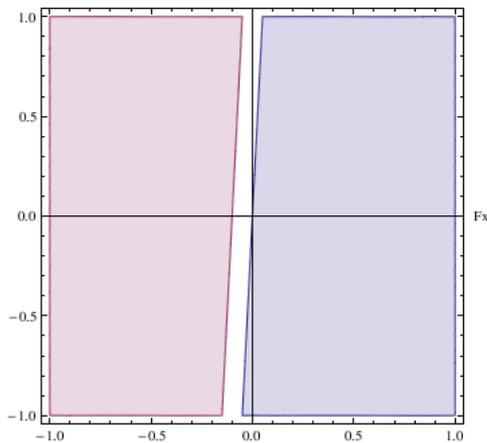
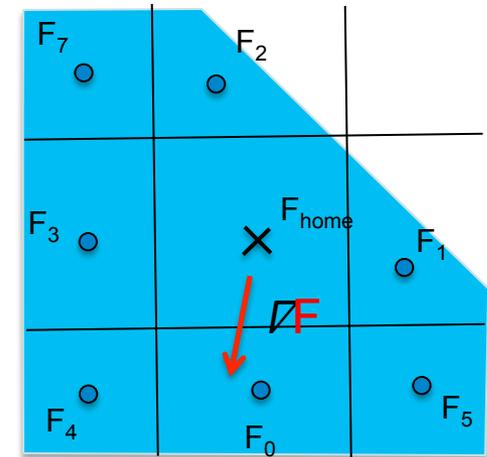
$$F_1 = F_{home} - \epsilon$$

$$\vec{X}_1 - \vec{X}_{home} = (\Delta x)\hat{i} + (\Delta y - \delta)\hat{j}$$

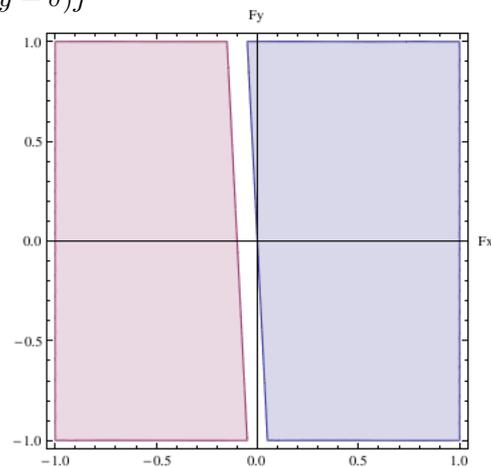
$$F_3 \geq F_{ext} \geq F_{home}$$

$$F_3 = F_{home} + \epsilon$$

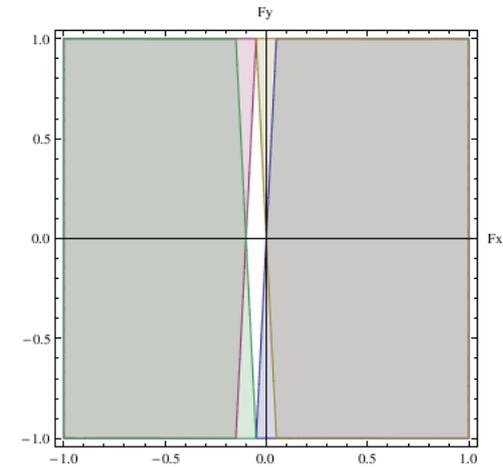
$$\vec{X}_3 - \vec{X}_{home} = -(\Delta x)\hat{i} + (\Delta y - \delta)\hat{j}$$



Ngr1 constraints



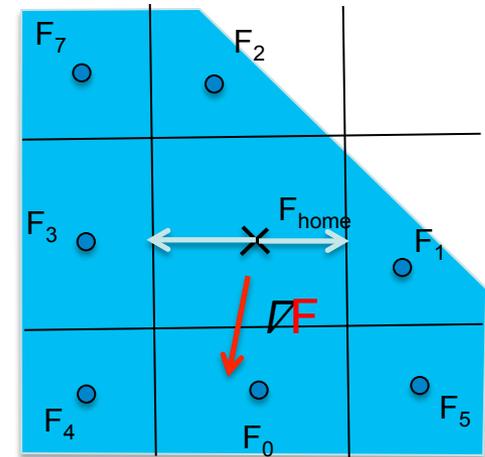
Ngr3 constraints



combined constraints

# Displacements used in constraints

- Excessive limiting is caused by opposite constraints *'folding into'* the gradient and hessian.
- To minimize this tendency: use displacement vector from home centroid to mid-edge coordinate or nodal coordinate. (remember these coordinates are evaluated after the Lagrangian step. Element is probably in a deformed configuration.)
- To minimize excessive limiting we *'diagonalize'* the displacements of opposite neighbors (define opposite displacements by vector between neighbors)



$$\hat{d} = \frac{\vec{X}_1 - \vec{X}_3}{|\vec{X}_1 - \vec{X}_3|}$$
$$\vec{d}_1 = |\vec{X}_1 - \vec{X}_c| \hat{d}$$
$$\vec{d}_3 = -|\vec{X}_3 - \vec{X}_c| \hat{d}$$

# Displacements used in constraints

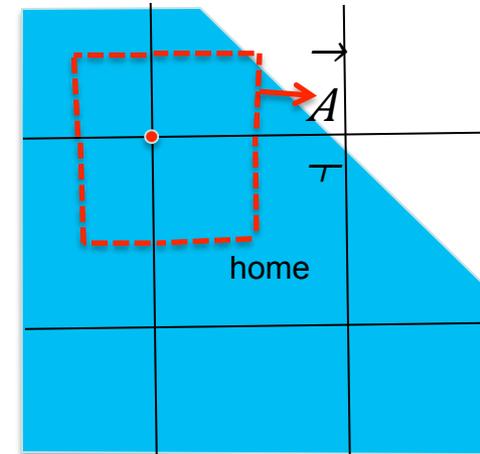
- Observation: constraints tend to be overly restrictive in many situations. Distributions will revert to 1<sup>st</sup> order when a monotonic 2<sup>nd</sup> or 3<sup>rd</sup> order distribution is visibly feasible
- Observation: gradient values tend to be lower than observed *average* gradient and Hessian values tend to be larger

# Additional 3<sup>rd</sup> order constraint

- Require that the gradient at the center of home element dotted with the gradient at each neighbor is  $\geq 0$ 
  - This ensures that a maxima or minima doesn't occur between home centroid and neighbor
- Constraint:
$$\text{Grad}(F_n) \cdot \text{Grad}(F_{\text{home}}) \geq 0$$
- Where:
$$\text{Grad}(F_n) = (F_x + dx F_{xx} + dy F_{xy}) i + (F_y + dy F_{yy} + dx F_{xy}) j$$
- To preserve the linearity of the constraint we evaluate an average gradient and use it:
$$\mathbf{F_x} F_x + \mathbf{F_y} F_y + dx \mathbf{F_x} F_{xx} + (dx \mathbf{F_y} + dy \mathbf{F_x}) F_{xy} + dy \mathbf{F_y} F_{yy} \geq 0$$

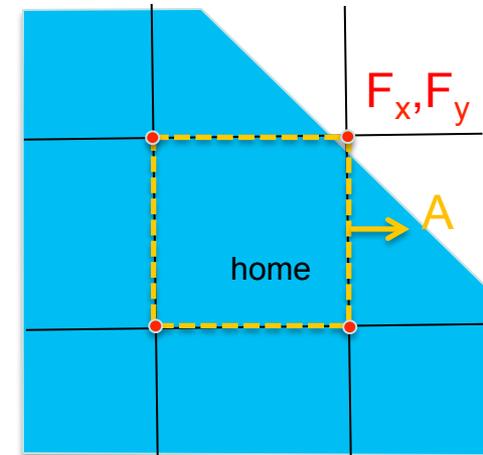
# Calculating an average gradient

- Use Dukowicz and Kodis' surface integral algorithm
- Accurate for distorted meshes
- Start with gradient around each home node
- Each line segment extends from element centroid to an edge midpoint
- $\text{Grad}(F) = \text{Sum}(F * \text{segment area}) / \text{volume inside polygon}$
- $\text{Avg Grad}(F) = \text{Sum}(\text{nodal Grad}(F) * \text{nodal volume}) / \text{Sum}(\text{nodal volume})$



# Calculating an average hessian

- Use Dukowicz and Kodis' surface integral algorithm again
- Compute the gradient of each component of the nodal gradients
- $F_{xx} = \text{grad}(F_x) * i$
- $F_{yy} = \text{grad}(F_y) * j$
- $F_{xy} = \text{grad}(F_x) * i + \text{grad}(F_y) * j$



# Final monotonicity constraint

- Magnitudes of gradient and hessian can be too large
- $\mathbf{F}_x \mathbf{F}_x + \mathbf{F}_y \mathbf{F}_y \leq \mathbf{F}_x \mathbf{F}_x + \mathbf{F}_x \mathbf{F}_x$
- $\mathbf{F}_{xx} \mathbf{F}_{xx} + \mathbf{F}_{xy} \mathbf{F}_{xy} + \mathbf{F}_{yy} \mathbf{F}_{yy} \leq \mathbf{F}_{xx} \mathbf{F}_{xx} + \mathbf{F}_{xy} \mathbf{F}_{xy} + \mathbf{F}_{yy} \mathbf{F}_{yy}$

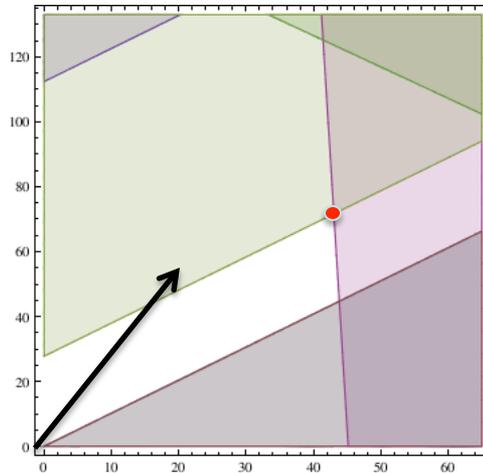
Where bold italics derivatives are the components of the average gradient and hessian

# Objective function

- A linear function which measures how close the extrapolated value is to each neighbor value
- Each occupied neighbor contributes to the objective function:  
If  $F_{\text{home}} \geq F_n$   
Obj +=  $F_{\text{ext}} - F_n$   
    +=  $F_{\text{home}} + dx F_x + dy F_y + dx dx F_{xx} + dx dy F_{xy} + dy dy F_y - F_n$   
    +=  $dx F_x + dy F_y + dx dx F_{xx} + dx dy F_{xy} + dy dy F_{yy} + F_{\text{home}} - F_n$
- Due to monotonicity constraints,  $F_{\text{home}} \geq F_{\text{ext}} \geq F_n$ , this can't go negative  
If  $F_{\text{home}} < F_n$   
Obj +=  $F_n - F_{\text{ext}}$   
    +=  $F_n - F_{\text{home}} - dx F_x - dy F_y - dx dx F_{xx} - dx dy F_{xy} - dy dy F_y$   
    -=  $dx F_x + dy F_y + dx dx F_{xx} + dx dy F_{xy} + dy dy F_{yy} + F_{\text{home}} - F_n$
- Constants are dropped
- Each neighbor's objective contribution is weighted by  $\text{abs}(F_{\text{home}} - F_n)$
- Minimize the objective
- Alternate objective functions

# Linear programming formulation

- The constraints and the objective function must be linear in the unknowns
- The null solution ( $F_x=F_y=F_{xx}=F_{xy}=F_{yy}=0$ ) must be a valid solution
- Allow negative solutions (ie  $F_x < 0$ ) by describing each unknown by two variables:  $F_x = F_x^+ - F_x^-$
- Only one of the two variables will be non-zero.
- For a 3<sup>rd</sup> order 2D element with 8 occupied neighbors:
  - There will be 1 objective function
  - There will be 10 unknowns
  - There will be 3 constraints per neighbor (24 neighbor constraints)
  - There will be 2 global constraints (magnitude of grad and magnitude of hessian)

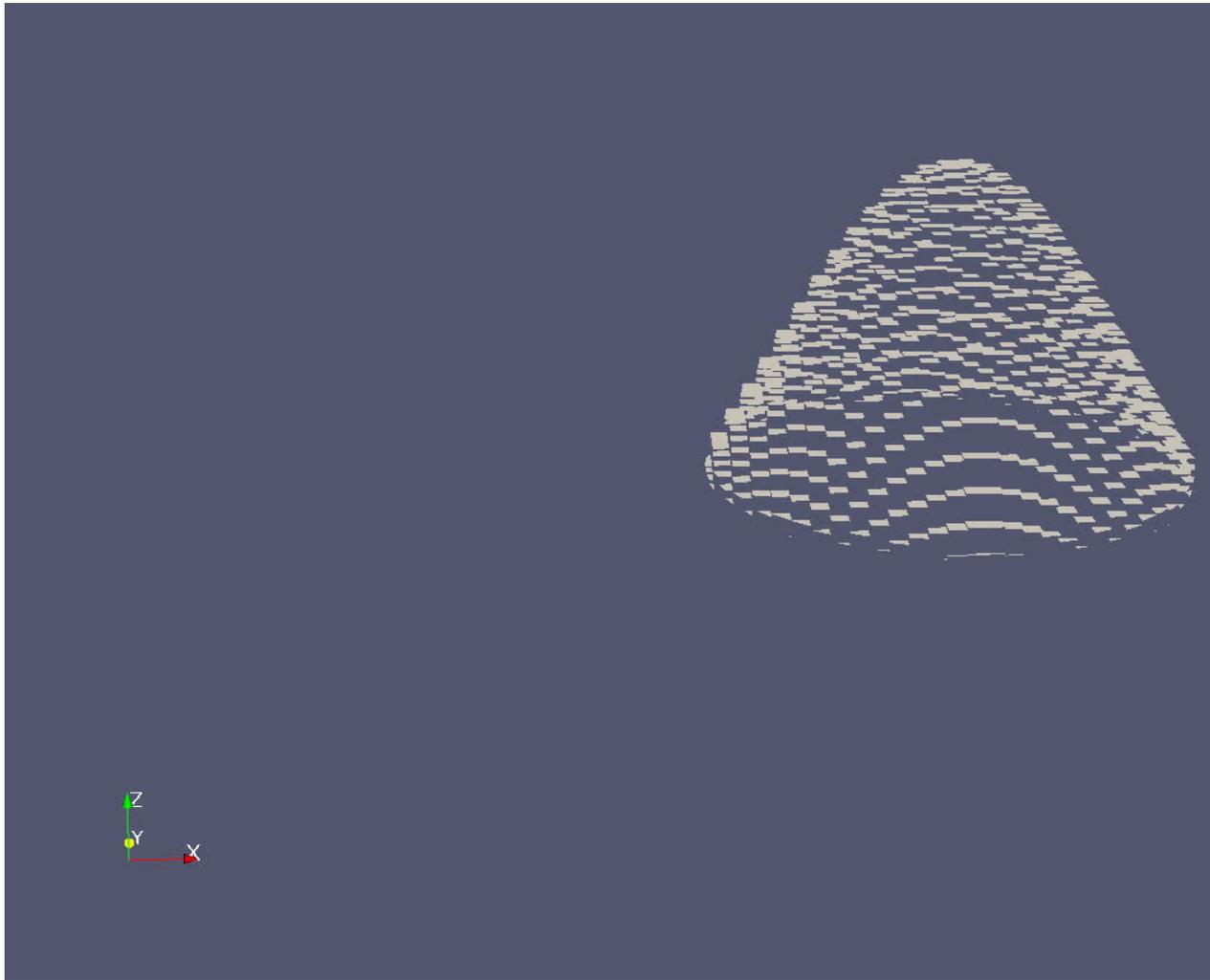


- For 2<sup>nd</sup> order distributions, there are no conservation terms
- For 3<sup>rd</sup> order distributions, after having determined the grad and hessian, we scale the distribution to conserve. For example, the density of an element times the material volume must equal the integral of the 3<sup>rd</sup> order distribution over the donor material polygon:
- $V_{xx} = [(1/2) \text{ volume integral}( dx dx ) \text{ over material polygon}]/\text{mat volume}$
- $V_{xy} = [\text{volume integral}( dx dy ) \text{ over material polygon } ]/\text{mat volume}$
- $V_{yy} = [(1/2) \text{ volume integral}( dy dy ) \text{ over material polygon } ]/\text{mat volume}$
- This distribution is integrated over each intersection polygon:
- $\Delta F = F_{\text{home}} - (F_{xx}(V_{xx}) + F_{xy}(V_{xy}) + F_{yy}(V_{yy}))\text{Volint} +$   
 $F_x \text{ volume integral}(dx) + F_y \text{ volume integral}(dy) +$   
 $F_{xx} \text{ volume integral}(dx dx/2) + F_{xy} \text{ volume integral}(dx dy) +$   
 $F_{yy} \text{ volume integral}(dy dy/2)$
- Several quantities are usually remapped; hence, evaluation and storage of intersection polygon integrals is economical

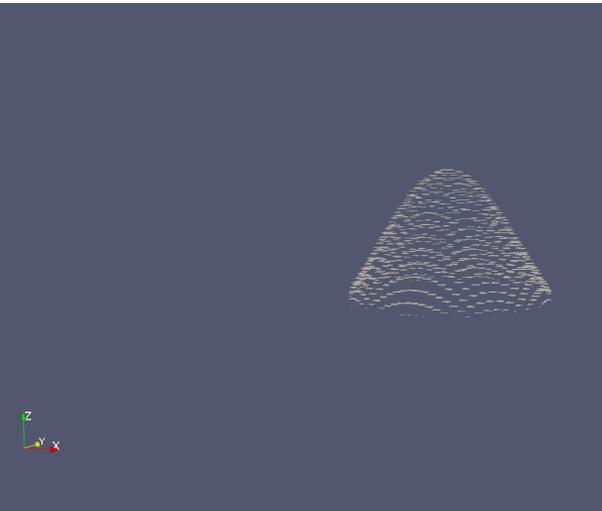
# Verification problem

- Rider's 2D distorting circle with a cosine shaped density distribution and a prescribed velocity
- The problem starts undeformed, it moves and distorts through 90 degrees and then reverses to return to its initial configuration
- Perfect solution would have no difference between starting and ending density distribution
- Show movies of 1<sup>st</sup> order, 2<sup>nd</sup> order, and 3<sup>rd</sup> order

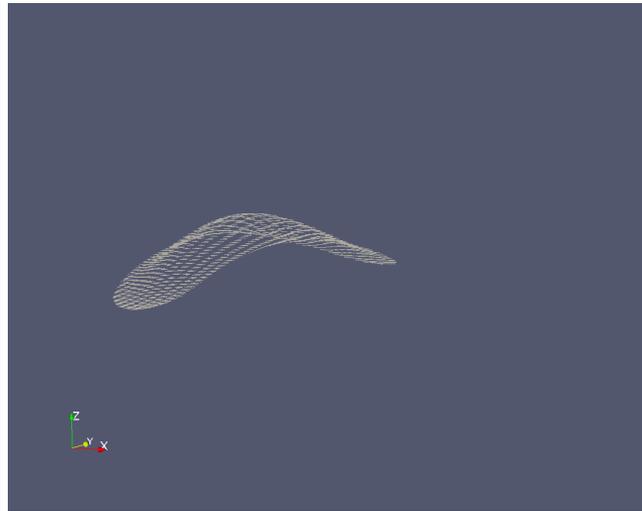
# Verification problem 1<sup>st</sup> order 144 elements



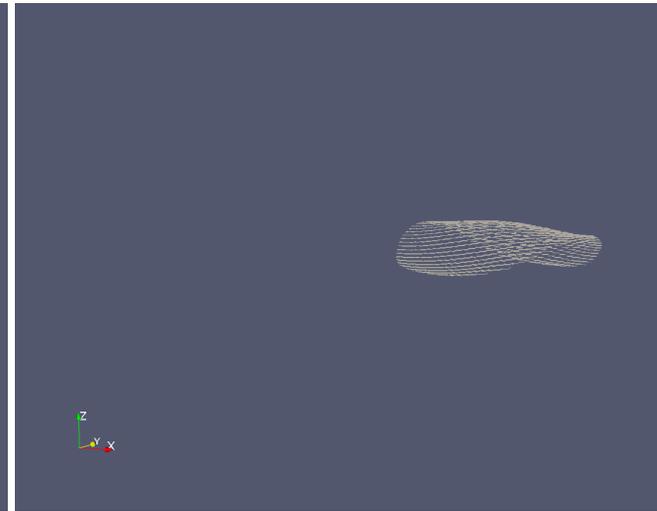
# Verification problem 1<sup>st</sup> order 144 elements



Start time

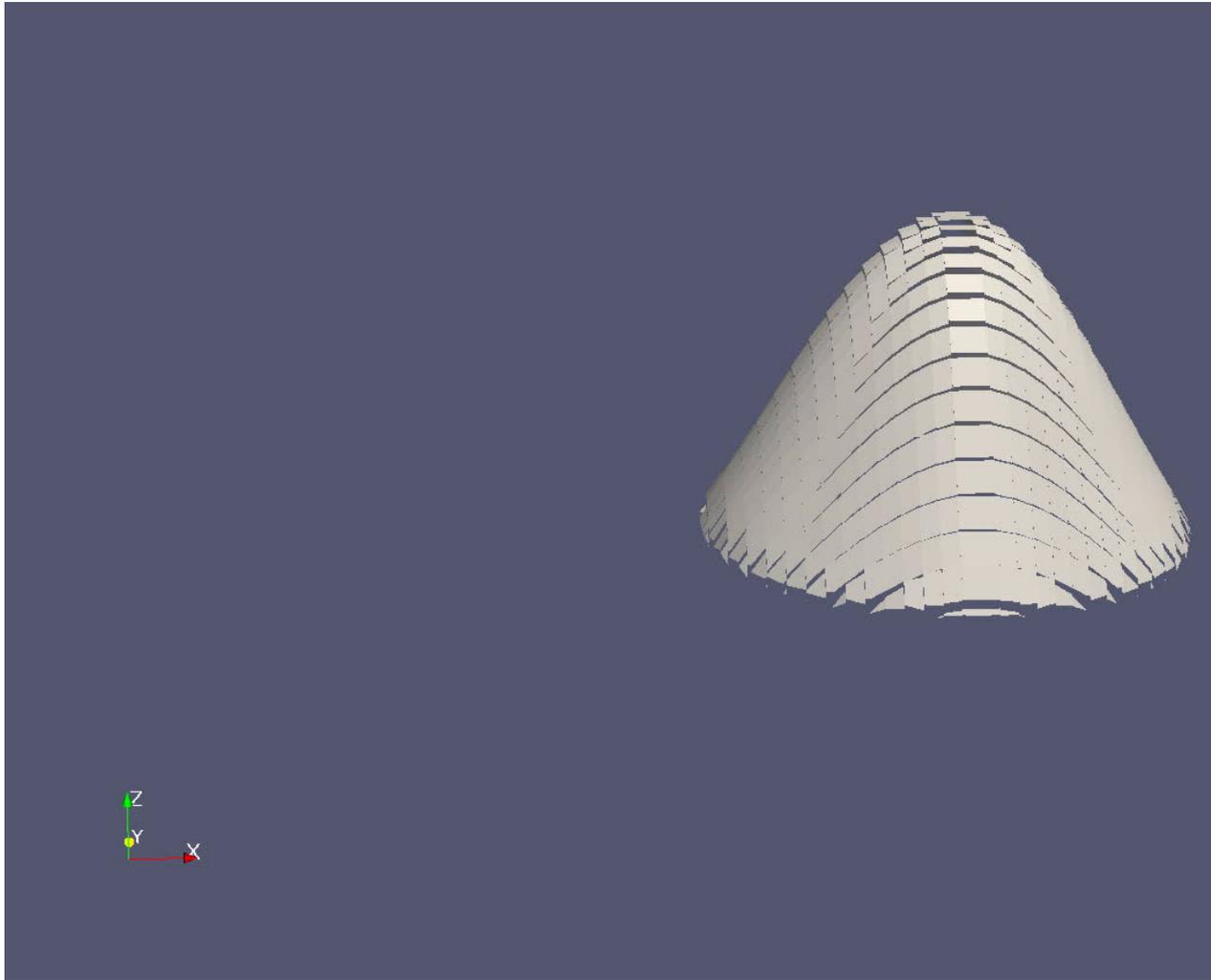


Reverse direction

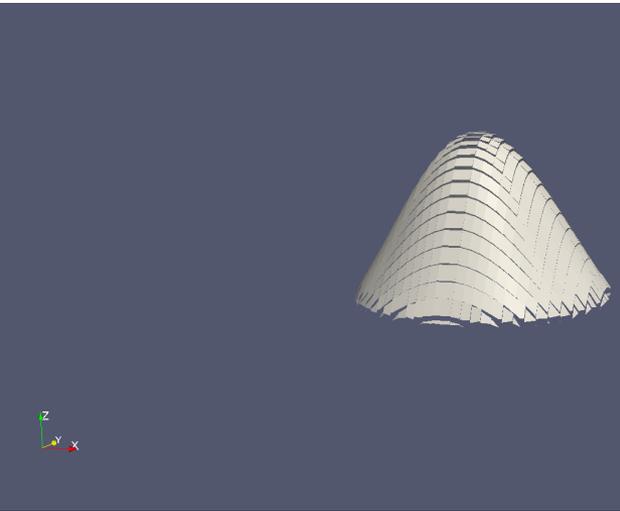


End time

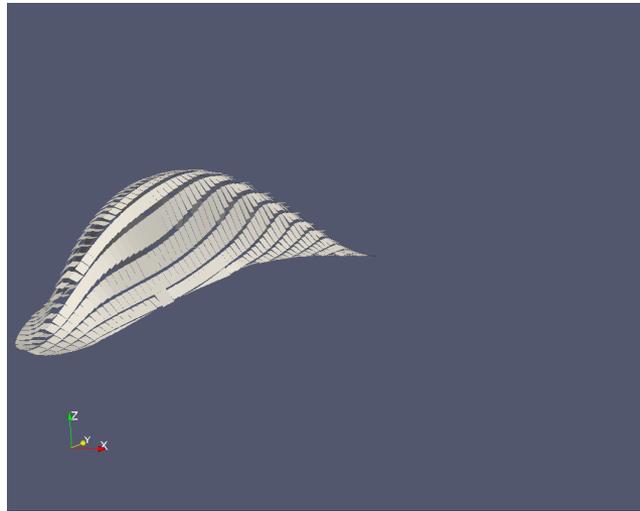
# Verification problem 2<sup>nd</sup> order 144 elements



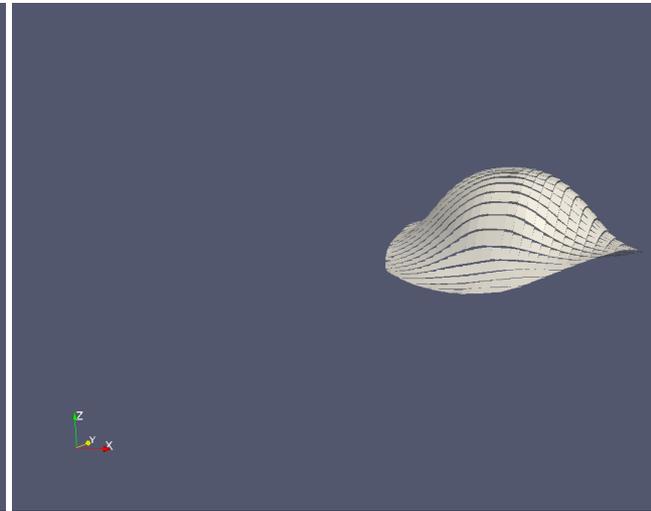
# Verification problem 2<sup>nd</sup> order 144 elements



Start time

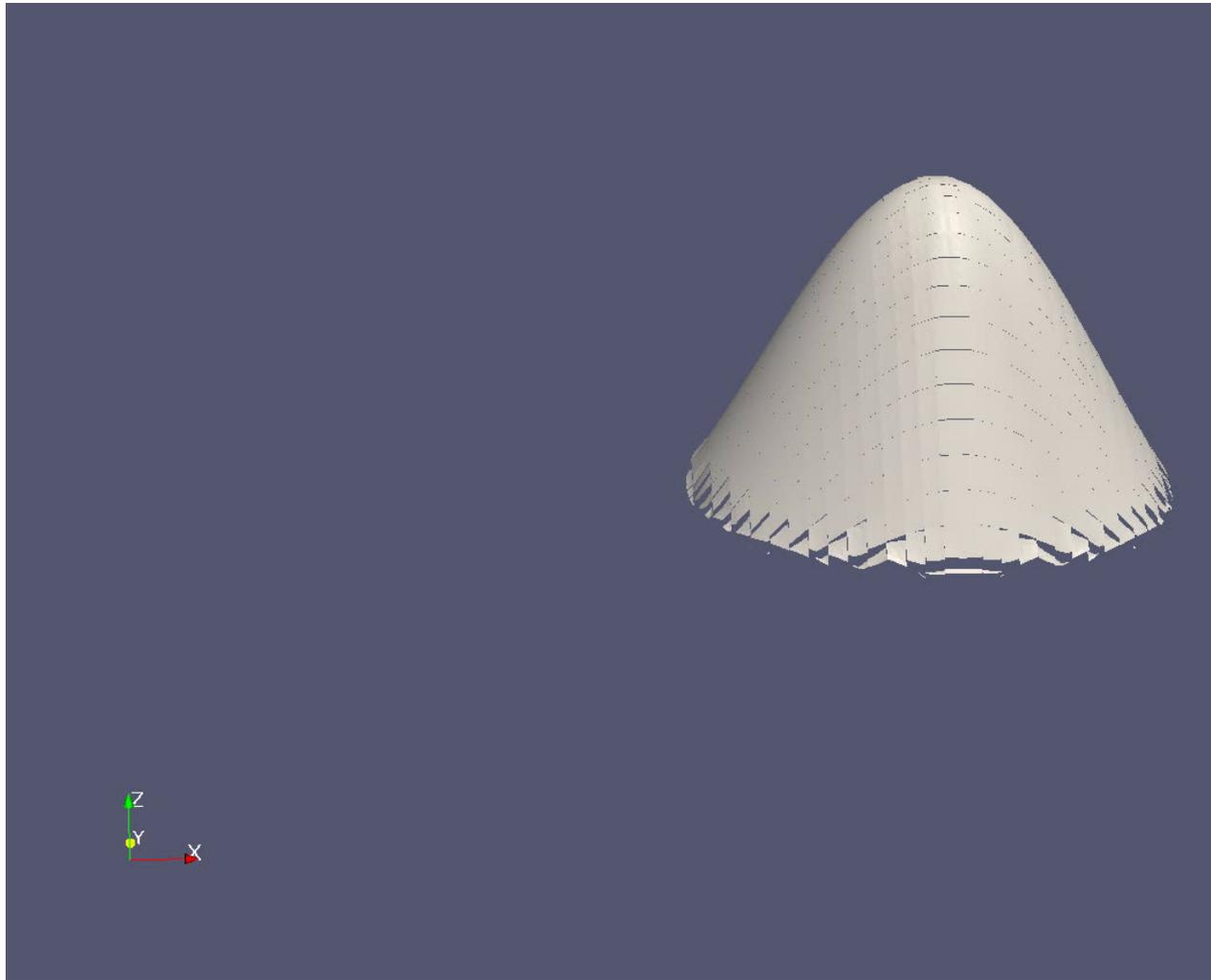


Reverse direction

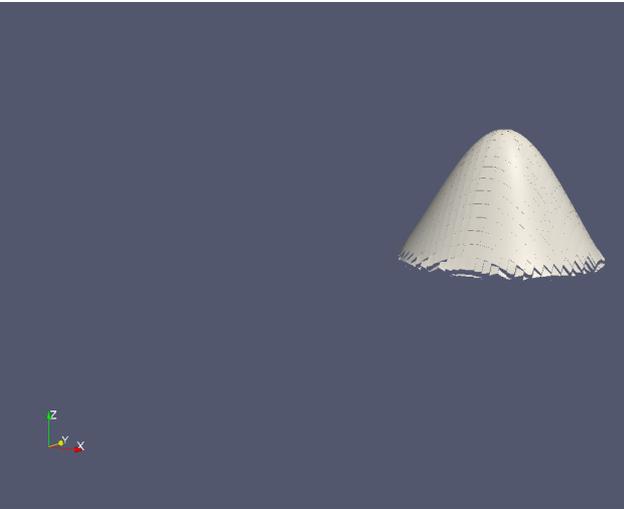


End time

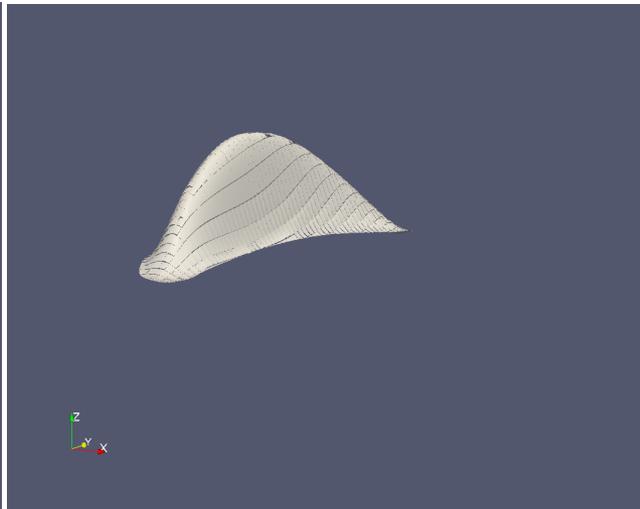
# Verification problem 3<sup>rd</sup> order 144 elements



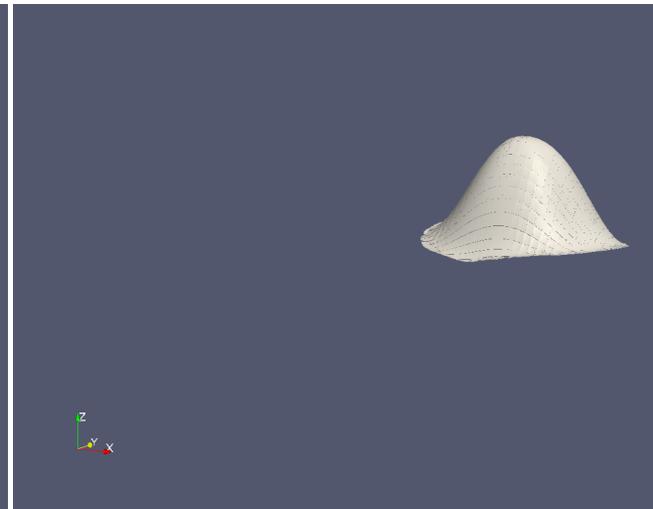
# Verification problem 3<sup>rd</sup> order 144 elements



Start time



Reverse direction



End time

- Demonstrated a variable (1<sup>st</sup>, 2<sup>nd</sup>, and 3<sup>rd</sup> order), monotonic, conservative, multi-dimensional remapping algorithm
- Algorithm is still under development and testing
- Compare multi-dimensional results with standard one-dimensional interpolation for alternating direction, swept volume remap